Unmasking Fault Tolerance: Masking vs. Non-masking Fault-tolerant Systems

Nils Müllner
nils.muellner@informatik.uni-oldenburg.de

System Software and Distributed Systems Group,
Universität Oldenburg, Germany

June 30, 2009
Table of contents

1 Who am I?
2 Unmasking Fault Tolerance
3 Simulation, Analysis and the Real World
4 Results So Far
5 Work In Progress
6 Conclusion
Automatic Verification and Analysis of Complex Systems

- transregional: 3 Universities, 3 * 4 years, now in year 6
- Oldenburg, Saarbrücken, Freiburg
- hybrid systems
- fault model driven stochastic model checking
- 12 supervisors, 19 postdocs, 54 students (18 finished)
Automatic Verification and Analysis of Complex Systems
transregional: 3 Universities, 3 * 4 years, now in year 6
Oldenburg, Saarbrücken, Freiburg
hybrid systems
fault model driven stochastic model checking
12 supervisors, 19 postdocs, 54 students (18 finished)
Automatic Verification and Analysis of Complex Systems

transregional: 3 Universities, 3 * 4 years, now in year 6

Oldenburg, Saarbrücken, Freiburg

- hybrid systems
- fault model driven stochastic model checking
- 12 supervisors, 19 postdocs, 54 students (18 finished)
Automatic Verification and Analysis of Complex Systems

transregional: 3 Universities, 3 * 4 years, now in year 6

Oldenburg, Saarbrücken, Freiburg

hybrid systems

fault model driven stochastic model checking

12 supervisors, 19 postdocs, 54 students (18 finished)
Automatic Verification and Analysis of Complex Systems

transregional: 3 Universities, $3 \times 4$ years, now in year 6

Oldenburg, Saarbrücken, Freiburg

hybrid systems

fault model driven stochastic model checking

12 supervisors, 19 postdocs, 54 students (18 finished)
Automatic Verification and Analysis of Complex Systems
transregional: 3 Universities, 3 * 4 years, now in year 6
Oldenburg, Saarbrücken, Freiburg
hybrid systems
fault model driven stochastic model checking
12 supervisors, 19 postdocs, 54 students (18 finished)
Graduate School funded by Deutsche Forschungsgemeinschaft DFG (German Research Foundation)

- Fault Tolerance Metrics = Availability, Reliability
- founded April 1st 2005
- 13 supervisors, 8 students (8 finished, 14 scholarships)
Graduate School funded by Deutsche Forschungsgemeinschaft DFG (German Research Foundation)

Fault Tolerance Metrics = Availability, Reliability

founded April 1st 2005

13 supervisors, 8 students (8 finished, 14 scholarships)
Graduate School funded by Deutsche Forschungsgemeinschaft DFG (German Research Foundation)

Fault Tolerance Metrics = Availability, Reliability

founded April 1st 2005

13 supervisors, 8 students (8 finished, 14 scholarships)
Graduate School funded by Deutsche Forschungsgemeinschaft DFG (German Research Foundation)

Fault Tolerance Metrics = Availability, Reliability

founded April 1st 2005

13 supervisors, 8 students (8 finished, 14 scholarships)
Thesis Topics

- **MSc**: Simulation of Self-Stabilizing Distributed Algorithms to Determine Fault Tolerance Measures
  - distributed systems
  - self-stabilizing algorithms (breadth-/depth-first search, mutual exclusion, leader election)
  - transient faults

- **PhD**: Unmasking Fault Tolerance
Thesis Topics

**MSc: Simulation of Self-Stabilizing Distributed Algorithms to Determine Fault Tolerance Measures**
- distributed systems
- self-stabilizing algorithms (breadth-/depth-first search, mutual exclusion, leader election)
- transient faults

**PhD: Unmasking Fault Tolerance**
MSc: Simulation of Self-Stabilizing Distributed Algorithms to Determine Fault Tolerance Measures
- distributed systems
- self-stabilizing algorithms (breadth-/depth-first search, mutual exclusion, leader election)
- transient faults

PhD: Unmasking Fault Tolerance
Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

<table>
<thead>
<tr>
<th></th>
<th>safe</th>
<th>(\neg) safe</th>
</tr>
</thead>
<tbody>
<tr>
<td>live</td>
<td>masking</td>
<td>nonmasking</td>
</tr>
<tr>
<td>(\neg) live</td>
<td>fail safe</td>
<td>-</td>
</tr>
</tbody>
</table>

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?
Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

<table>
<thead>
<tr>
<th></th>
<th>safe</th>
<th>¬ safe</th>
</tr>
</thead>
<tbody>
<tr>
<td>live</td>
<td>masking</td>
<td>nonmasking</td>
</tr>
<tr>
<td>¬ live</td>
<td>fail safe</td>
<td>-</td>
</tr>
</tbody>
</table>

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?
Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

<table>
<thead>
<tr>
<th></th>
<th>safe</th>
<th>¬ safe</th>
</tr>
</thead>
<tbody>
<tr>
<td>live</td>
<td>masking</td>
<td>nonmasking</td>
</tr>
<tr>
<td>¬ live</td>
<td>fail safe</td>
<td>-</td>
</tr>
</tbody>
</table>

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?
Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

<table>
<thead>
<tr>
<th>safe</th>
<th>¬ safe</th>
</tr>
</thead>
<tbody>
<tr>
<td>live</td>
<td>masking</td>
</tr>
<tr>
<td>¬ live</td>
<td>fail safe</td>
</tr>
</tbody>
</table>
Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

<table>
<thead>
<tr>
<th></th>
<th>safe</th>
<th>¬ safe</th>
</tr>
</thead>
<tbody>
<tr>
<td>live</td>
<td>masking</td>
<td>nonmasking</td>
</tr>
<tr>
<td>¬ live</td>
<td>fail safe</td>
<td>-</td>
</tr>
</tbody>
</table>

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?
Motivation

- focusing on distributed systems
- fault tolerance: reliability, availability, maintainability, ...
- it is a quality of service

<table>
<thead>
<tr>
<th></th>
<th>safe</th>
<th>¬ safe</th>
</tr>
</thead>
<tbody>
<tr>
<td>live</td>
<td>masking</td>
<td>nonmasking</td>
</tr>
<tr>
<td>¬ live</td>
<td>fail safe</td>
<td>-</td>
</tr>
</tbody>
</table>

- nonmasking fault tolerance exposes user to faults
- masking fault tolerance conceals faults
- but at what cost?
Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

- tradeoff between degree of masking vs. cost not investigated, yet

- how much time/space per "quality of service"?
Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

- tradeoff between degree of masking vs. cost not investigated, yet

- how much time/space per "quality of service"?
Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

- tradeoff between degree of masking vs. cost not investigated, yet

- how much time/space per "quality of service"?
Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

- tradeoff between degree of masking vs. cost not investigated, yet

- how much time/space per "quality of service"?
Motivation

- cost of masking: redundancy
  - spatial: coding, TMR, soft-/hardware
  - temporal: retransmission, multiplexing

- tradeoff between degree of masking vs. cost not investigated, yet

- how much time/space per “quality of service”?
Scientific Triangle

Real World Experiments

Simulation

Analysis
Methods for Derivation

- formal analysis $\Rightarrow$ state space explosion
- simulation $\Rightarrow$ inaccurate, no proof
- real world experiments $\Rightarrow$ expensive
Methods for Derivation

- formal analysis $\Rightarrow$ state space explosion
- simulation $\Rightarrow$ inaccurate, no proof
- real world experiments $\Rightarrow$ expensive
Methods for Derivation

- formal analysis $\Rightarrow$ state space explosion
- simulation $\Rightarrow$ inaccurate, no proof
- real world experiments $\Rightarrow$ expensive
Methods for Derivation

- formal analysis \implies state space explosion
- simulation \implies inaccurate, no proof
- real world experiments \implies expensive
Methods for Derivation

- formal analysis $\Rightarrow$ state space explosion
- simulation $\Rightarrow$ inaccurate, no proof
- real world experiments $\Rightarrow$ expensive
Methods for Derivation

- formal analysis $\Rightarrow$ state space explosion
- simulation $\Rightarrow$ inaccurate, no proof
- real world experiments $\Rightarrow$ expensive
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
  - probabilistic model checker PRISM:
    - generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow$~ 7 billion states)
    - checks model against predicates
    - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
    - can express reliability & availability
  - not parallelized (uses only one core per instance)
  - potential for further optimization
  - highly precise
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):

  - probabilistic model checker PRISM:
    - generates model (3 proc ⇒ 6561 states, 5 proc ⇒ ~ 7 billion states)
    - checks model against predicates
    - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
    - can express reliability & availability

- not parallelized (uses only one core per instance)

- potential for further optimization

- highly precise
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):

- probabilistic model checker PRISM:
  - generates model (3 proc $\Rightarrow 6561$ states, 5 proc $\Rightarrow \sim 7$ billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., $0.0756312876123$ is the steady state probability of state $x$)
  - can express reliability & availability

- not parallelized (uses only one core per instance)

- potential for further optimization

- highly precise
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
  - probabilistic model checker PRISM:
    - generates model (3 proc ⇒ 6561 states, 5 proc ⇒ ~ 7 billion states)
    - checks model against predicates
    - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
    - can express reliability & availability

- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise

Nils Müllner (Universität Oldenburg)
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
  - probabilistic model checker PRISM:
    - generates model (3 proc $\Rightarrow 6561$ states, 5 proc $\Rightarrow \sim 7$ billion states)
    - checks model against predicates
    - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state $x$)
    - can express reliability & availability
  
- not parallelized (uses only one core per instance)

- potential for further optimization

- highly precise
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
  - probabilistic model checker PRISM:
    - generates model (3 proc ⇒ 6561 states, 5 proc ⇒∼ 7 billion states)
    - checks model against predicates
    - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
    - can express reliability & availability

- not parallelized (uses only one core per instance)
- potential for further optimization
- highly precise
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):

- probabilistic model checker PRISM:
  - generates model (3 proc ⇒ 6561 states, 5 proc ⇒ ~ 7 billion states)
  - checks model against predicates
  - gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
  - can express reliability & availability

- not parallelized (uses only one core per instance)

- potential for further optimization

- highly precise
Formal Analysis

- Based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
  - Probabilistic model checker PRISM:
    - Generates model (3 proc $\Rightarrow$ 6561 states, 5 proc $\Rightarrow$ $\sim$ 7 billion states)
    - Checks model against predicates
    - Gives quantitative answers (e.g., 0.0756312876123 is the steady state probability of state x)
    - Can express reliability & availability

- Not parallelized (uses only one core per instance)

- Potential for further optimization

- Highly precise
Formal Analysis

- based on Markov models (DTMC, CTMC, DTMDP, CTMDP):
  - probabilistic model checker PRISM:
    - generates model (3 proc $\Rightarrow 6561$ states, 5 proc $\Rightarrow \sim 7$ billion states)
    - checks model against predicates
    - gives quantitative answers (e.g., $0.0756312876123$ is the steady state probability of state $x$)
    - can express reliability & availability

- not parallelized (uses only one core per instance)

- potential for further optimization

- highly precise
based on simulation framework *SiSSDA* (Simulator for Self-Stabilizing Distributed Algorithms)

- derives approximate results fast and refines them over time
- delivers no proof
- runs parallelized, utilizing up to \( n + 2 \) cores
  (nodes + fault-injector + server)
- based on simulation framework *SiSSDA* (Simulator for Self-Stabilizing Distributed Algorithms)
- derives approximate results fast and refines them over time
- delivers no proof
- runs parallelized, utilizing up to $n + 2$ cores (nodes + fault-injector + server)
based on simulation framework SiSSDA  
(Simulator for Self-Stabilizing Distributed Algorithms)  
derives approximate results fast and refines them over time  
delivers no proof  
runs parallelized, utilizing up to $n + 2$ cores  
(nodes + fault-injector + server)
based on simulation framework SiSSDA (Simulator for Self-Stabilizing Distributed Algorithms)
- derives approximate results fast and refines them over time
- delivers no proof
- runs parallelized, utilizing up to \( n + 2 \) cores (nodes + fault-injector + server)
Real World Experiments

- applicable scenarios: wireless sensor networks (WSN)
  - too expensive (≈200 AUD/mote, ≈100 motes for a realistic network)
  - too time intensive (program and place 100 motes, charge 100 batteries, measure and evaluate)
  - hence, derivation of good solutions by analysis & simulation before the testing in real world should save money/time
Real World Experiments

- applicable scenarios: wireless sensor networks (WSN)
  - too expensive ($\sim 200$ AUD/mote, $\sim 100$ motes for a realistic network)
  - too time intensive (program and place 100 motes, charge 100 batteries, measure and evaluate)
  - hence, derivation of good solutions by analysis & simulation before the testing in real world should save money/time
Real World Experiments

- applicable scenarios: wireless sensor networks (WSN)
  - too expensive ($\sim$200 AUD/mote, $\sim$100 motes for a realistic network)
  - too time intensive (program and place 100 motes, charge 100 batteries, measure and evaluate)

  hence, derivation of good solutions by analysis & simulation before the testing in real world should save money/time
Real World Experiments

- applicable scenarios: wireless sensor networks (WSN)
  
  - too expensive (\(\sim 200\) AUD/mote, \(\sim 100\) motes for a realistic network)
  
  - too time intensive (program and place 100 motes, charge 100 batteries, measure and evaluate)
  
  - hence, derivation of good solutions by analysis & simulation before the testing in real world should save money/time
Combining the Three Approaches

- build small scenarios for analysis and simulation
  
- verify simulation results by analysis for small scenarios
  
- after simulation & analysis proved fine for small systems, optimize large scenarios based on simulation results
  
- result: manageable set of close-to-optimal solutions
  
- then test given results in real world for real measures
Combining the Three Approaches

- build small scenarios for analysis and simulation

- verify simulation results by analysis for small scenarios

  - after simulation & analysis proved fine for small systems, optimize large scenarios based on simulation results

  - result: manageable set of close-to-optimal solutions

  - then test given results in real world for real measures
Combining the Three Approaches

- build small scenarios for analysis and simulation
- verify simulation results by analysis for small scenarios
- after simulation & analysis proved fine for small systems, optimize large scenarios based on simulation results
  - result: manageable set of close-to-optimal solutions
  - then test given results in real world for real measures
Combining the Three Approaches

- build small scenarios for analysis and simulation
- verify simulation results by analysis for small scenarios
- after simulation & analysis proved fine for small systems, optimize large scenarios based on simulation results
- result: manageable set of close-to-optimal solutions
- then test given results in real world for real measures
Results So Far

- Simulation and analysis deliver similar results (two papers)

- Derivation of fault tolerance measures challenging
  - Accuracy of results vs computation time
  - System model definition must be extremely precise (e.g., communication model)

- Definition of new metrics (Instantaneous Window Availability)
Results So Far

- simulation and analysis deliver similar results (two papers)

- derivation of fault tolerance measures challenging
  - accuracy of results vs computation time
  - system model definition must be extremely precise (e.g., communication model)

- definition of new metrics (Instantaneous Window Availability)
Results So Far

- simulation and analysis deliver similar results (two papers)

- derivation of fault tolerance measures challenging
  - accuracy of results vs computation time
  - system model definition must be extremely precise (e.g., communication model)

- definition of new metrics (Instantaneous Window Availability)
Results So Far

- simulation and analysis deliver similar results (two papers)

- derivation of fault tolerance measures challenging
  - accuracy of results vs computation time
  - system model definition must be extremely precise (e.g., communication model)

- definition of new metrics (Instantaneous Window Availability)
Results So Far

- Simulation and analysis deliver similar results (two papers)

- Derivation of fault tolerance measures challenging
  - Accuracy of results vs computation time
  - System model definition must be extremely precise (e.g., communication model)

- Definition of new metrics (Instantaneous Window Availability)
Instantaneous Window Availability

- for measuring relation/tradeoff between cost (time) and degree of masking

- to be extended: IW reliability, maintainability,...

- to be extended: relation of cost (space) and degree of masking
Instantaneous Window Availability

- for measuring relation/tradeoff between cost (time) and degree of masking

- to be extended: IW reliability, maintainability,...

- to be extended: relation of cost (space) and degree of masking
Instantaneous Window Availability

- for measuring relation/tradeoff between cost (time) and degree of masking

- to be extended: IW reliability, maintainability, ...

- to be extended: relation of cost (space) and degree of masking
Instantaneous Window Availability

(a) (b)

Nils Müllner (Universität Oldenburg) Unmasking Fault Tolerance 16 / 25
- automatic computation of Pareto-optimal solution sets

- derivation of trade-off between availability, consistency and redundancy (space) for WSNs
- automatic computation of Pareto-optimal solution sets

```
MoOP -> FEMKE
  |   |   |
  v   v   v
test sol. result
  |   |   |
  v   v   v
PRISM
```

- derivation of trade-off between availability, consistency and redundancy (space) for WSNs
**Conclusion**

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
  - problems are often too complex
    - to model
    - to analyze
  - simulation alleviates this burden somewhat at the cost of accuracy/proof

  Yet, satisfactory solutions can be derived in a reasonable amount of time, if exact calculation takes too long.
Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex to model or to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonable amount of time, if exact calculation takes too long
Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonable amount of time if exact calculation takes too long
Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex to model to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonable amount of time if exact calculation takes too long
Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof
- yet, satisfactory solutions can be derived in a reasonable amount of time, if exact calculation takes too long
Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world

- problems are often too complex
  - to model
  - to analyze

- simulation alleviates this burden somewhat at the cost of accuracy/proof

- yet, satisfactory solutions can be derived in a reasonable amount of time, if exact calculation takes too long
deriving a good trade-off between masking and nonmasking is not trivial

- state space explosion in analysis
- inaccurate results in simulation
- high costs in real world

problems are often too complex

- to model
- to analyze

simulation alleviates this burden somewhat at the cost of accuracy/proof

yet, satisfactory solutions can be derived in a reasonable amount of time, if exact calculation takes too long
Conclusion

- deriving a good trade-off between masking and nonmasking is not trivial
  - state space explosion in analysis
  - inaccurate results in simulation
  - high costs in real world
- problems are often too complex
  - to model
  - to analyze
- simulation alleviates this burden somewhat at the cost of accuracy/proof

yet, satisfactory solutions can be derived in a reasonable amount of time, if exact calculation takes too long.


- **Conclusion**

  - deriving a good trade-off between masking and nonmasking is not trivial
    - state space explosion in analysis
    - inaccurate results in simulation
    - high costs in real world

  - problems are often too complex
    - to model
    - to analyze

  - simulation alleviates this burden somewhat at the cost of accuracy/proof

  - yet, *satisfactory* solutions can be derived in a reasonable amount of time, if exact calculation takes too long


self-stabilizing systems are nonmasking fault tolerant

during stabilization/repair phase, user is exposed to faults

why not even weaken nonmasking self-stabilizing systems?

unmask fault tolerance even below nonmasking...
self-stabilizing systems are nonmasking fault tolerant

during stabilization/repair phase, user is exposed to faults

why not even weaken nonmasking self-stabilizing systems?

unmask fault tolerance even below nonmasking...
• self-stabilizing systems are nonmasking fault tolerant

• during stabilization/repair phase, user is exposed to faults

• why not even weaken nonmasking self-stabilizing systems?

• unmask fault tolerance even below nonmasking...
self-stabilizing systems are nonmasking fault tolerant

during stabilization/repair phase, user is exposed to faults

why not even weaken nonmasking self-stabilizing systems?

unmask fault tolerance even below nonmasking...
Bonus: fun with weakening the masking property 2/3

Unmasking Fault Tolerance

Nils Müllner (Universität Oldenburg)
• convergence: $\exists t_k : x(t_k) \models P$

• closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$

• attack at convergence: slow or probabilistic

• attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$
- convergence: $\exists t_k : x(t_k) \models P$

- closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$

- attack at convergence: slow or probabilistic

- attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$
• convergence: $\exists t_k : x(t_k) \models P$

• closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$

• attack at convergence: slow or probabilistic

• attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$
• convergence: \( \exists t_k : x(t_k) \models P \)

• closure: \( \forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P \)

• attack at convergence: slow or probabilistic

• attack at closure: \( \exists t_0 \forall t \geq t_0 : x(t) \models P \)
convergence: $\exists t_k : x(t_k) \models P$

closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$

attack at convergence: slow or probabilistic

attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$
convergence: $\exists t_k : x(t_k) \models P$

closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$

attack at convergence: slow or probabilistic

attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$
• convergence: $\exists t_k : x(t_k) \models P$

• closure: $\forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P$

• attack at convergence: slow or probabilistic

• attack at closure: $\exists t_0 \forall t \geq t_0 : x(t) \models P$
Bonus: fun with weakening the masking property 3/3

- convergence: \( \exists t_k : x(t_k) \models P \)

- closure: \( \forall t \geq t_k : x(t_k) \models P \Rightarrow x(t) \models P \)

- attack at convergence: slow or probabilistic

- attack at closure: \( \exists t_0 \forall t \geq t_0 : x(t) \models P \)
Figure: slower stabilization

Figure: probabilistic stabilization
Attack Closure

set ofillegal states → set of legal states → set of legal states

Nils Müller (Universität Oldenburg)
Thank your for your attention.

nils.muellner@informatik.uni-oldenburg.de

Questions?
Thank your for your attention.

nils.muellner@informatik.uni-oldenburg.de

Questions?
Thank your for your attention.

nils.muellner@informatik.uni-oldenburg.de

Questions?